12-10-99

Express Mail Label No. EL442683403US

A

UTILITY PATENT APPLICATION TRANSMITTAL **(Large Entity)** *(Only for new nonprovisional applications under 37 CFR 1.53(b))*	Docket No. 2204/185
	Total Pages in this Submission 22

TO THE ASSISTANT COMMISSIONER FOR PATENTS

Box Patent Application
Washington, D.C. 20231

Transmitted herewith for filing under 35 U.S.C. 111(a) and 37 C.F.R. 1.53(b) is a new utility patent application for an invention entitled:

EXPEDITING AN OPERATION IN A COMPUTER SYSTEM

and invented by:

Bradley Cain

If a CONTINUATION APPLICATION, *check appropriate box and supply the requisite information:*

□ **Continuation** □ **Divisional** □ **Continuation-in-part (CIP)** of prior application No.: ________

Which is a:

□ **Continuation** □ **Divisional** □ **Continuation-in-part (CIP)** of prior application No.: ________

Which is a:

□ **Continuation** □ **Divisional** □ **Continuation-in-part (CIP)** of prior application No.: ________

Enclosed are:

Application Elements

1. □ Filing fee as calculated and transmitted as described below

2. ☒ Specification having 11 pages and including the following:

   a. ☒ Descriptive Title of the Invention

   b. □ Cross References to Related Applications *(if applicable)*

   c. □ Statement Regarding Federally-sponsored Research/Development *(if applicable)*

   d. □ Reference to Microfiche Appendix *(if applicable)*

   e. ☒ Background of the Invention

   f. ☒ Brief Summary of the Invention

   g. ☒ Brief Description of the Drawings *(if drawings filed)*

   h. ☒ Detailed Description

   i. ☒ Claim(s) as Classified Below

   j. ☒ Abstract of the Disclosure

UTILITY PATENT APPLICATION TRANSMITTAL **(Large Entity)** *(Only for new nonprovisional applications under 37 CFR 1.53(b))*	Docket No. 2204/185
	Total Pages in this Submission 22

Application Elements (Continued)

3. ☒ Drawing(s) *(when necessary as prescribed by 35 USC 113)*

   a. ☐ Formal Number of Sheets ____________

   b. ☒ Informal Number of Sheets 3

4. ☒ Oath or Declaration

   a. ☐ Newly executed *(original or copy)* ☒ Unexecuted

   b. ☐ Copy from a prior application (37 CFR 1.63(d)) *(for continuation/divisional application only)*

   c. ☒ With Power of Attorney ☐ Without Power of Attorney

   d. ☐ *DELETION OF INVENTOR(S)*
   Signed statement attached deleting inventor(s) named in the prior application, see 37 C.F.R. 1.63(d)(2) and 1.33(b).

5. ☐ Incorporation By Reference *(usable if Box 4b is checked)*
   The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.

6. ☐ Computer Program in Microfiche *(Appendix)*

7. ☐ Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all must be included)*

   a. ☐ Paper Copy

   b. ☐ Computer Readable Copy *(identical to computer copy)*

   c. ☐ Statement Verifying Identical Paper and Computer Readable Copy

Accompanying Application Parts

8. ☐ Assignment Papers *(cover sheet & document(s))*

9. ☐ 37 CFR 3.73(B) Statement *(when there is an assignee)*

10. ☐ English Translation Document *(if applicable)*

11. ☐ Information Disclosure Statement/PTO-1449 ☐ Copies of IDS Citations

12. ☐ Preliminary Amendment

13. ☒ Acknowledgment postcard

14. ☒ Certificate of Mailing

    ☐ First Class ☒ Express Mail *(Specify Label No.)*: EL442683403US

UTILITY PATENT APPLICATION TRANSMITTAL
(Large Entity)

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Docket No.
2204/185

Total Pages in this Submission
22

Accompanying Application Parts (Continued)

15. □ Certified Copy of Priority Document(s) *(if foreign priority is claimed)*

16. □ Additional Enclosures *(please identify below)*:

Fee Calculation and Transmittal

CLAIMS AS FILED

For	#Filed	#Allowed	#Extra		Rate	Fee
Total Claims	15	- 20 =	0	x	$18.00	$0.00
Indep. Claims	3	- 3 =	0	x	$78.00	$0.00
Multiple Dependent Claims (check if applicable) □						$0.00
					BASIC FEE	$760.00
OTHER FEE *(specify purpose)*						$0.00
					TOTAL FILING FEE	$760.00

□ A check in the amount of to cover the filing fee is enclosed.

□ The Commissioner is hereby authorized to charge and credit Deposit Account No. as described below. A duplicate copy of this sheet is enclosed.

- □ Charge the amount of as filing fee.
- □ Credit any overpayment.
- □ Charge any additional filing fees required under 37 C.F.R. 1.16 and 1.17.
- □ Charge the issue fee set in 37 C.F.R. 1.18 at the mailing of the Notice of Allowance, pursuant to 37 C.F.R. 1.311(b).

Signature

Jeffrey T. Klayman, Reg. No. 39,250
BROMBERG & SUNSTEIN LLP
125 Summer Street
Boston, MA 02110
(617) 443-9292

Dated: December 9, 1999

cc:

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR UNITED STATES PATENT

FOR

EXPEDITING AN OPERATION IN A COMPUTER SYSTEM

Inventor:

Bradley Cain
295 Harvard Street #804
Cambridge, MA 02139

Attorney Docket: 2204/185 (BA409)

Attorneys:

BROMBERG & SUNSTEIN LLP
125 Summer Street
Boston, MA 02110
(617) 443-9292

EXPEDITING AN OPERATION IN A COMPUTER SYSTEM

FIELD OF THE INVENTION

The present invention relates generally to computer systems, and more particularly to expediting an operation in a computer system.

BACKGROUND OF THE INVENTION

In today's information age, communication networks are often used for interconnecting computers and computer peripherals. A communication network typically includes a number of nodes that interoperate to route protocol messages. The various nodes in the communication network utilize various routing protocols in order to determine the routes that are used to route the protocol messages. Thus, each node is a computer that performs, among other things, various routing protocols in order to route protocol messages.

One type of routing protocol, known as a "link state" routing protocol, determines routes based upon the status of communication links between the various nodes. A link state routing protocol, such as OSPF and IS-IS, requires each node to have complete topology information. Therefore, each node periodically tests the communication links to each of its neighbors and sends a link state advertisement (LSA) protocol message including the link status information to all of the other nodes. Each node computes the routes based upon the link status information received from the other nodes.

When a node receives a LSA protocol message, the node updates its topology information database to include the link status information received in the LSA protocol message, and runs a special algorithm to determine the routes based upon the updated topology information. One well-known algorithm for determining routes is the Dijkstra shortest path algorithm. The Dijkstra shortest path algorithm computes the shortest paths to all destinations from a single source.

In order for the nodes in the communication network to properly route protocol messages, it is important for each node to compute routes in a timely manner. Unfortunately, the Dijkstra shortest path algorithm is computationally intensive, and can take a relatively long time to complete, especially in communication networks having many nodes.

Thus, a technique for reducing the amount of time required to complete the Dijkstra shortest path computation is needed.

SUMMARY OF THE INVENTION

In accordance with one aspect of the invention, logic for performing an operation is included in an operating system task. The operating system task typically executes at a low priority level. However, when the operation needs to be performed, the operating system task is raised to a high priority level, and the operation is performed at the high priority level. In this way, the operation is not interrupted by other operating system tasks that execute below the high priority level.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects and advantages of the invention will be appreciated more fully from the following further description thereof with reference to the accompanying drawings wherein:

FIG. 1 is a block diagram showing the relevant components of an exemplary node in accordance with an embodiment of the invention;

FIG. 2 is a logic flow diagram showing exemplary task priority control logic for expediting the Dijkstra shortest path computation in accordance with an embodiment of the invention; and

FIG. 3 is a logic flow diagram showing exemplary task priority control logic for expediting a computation in accordance with an embodiment of the invention.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

In an embodiment of the invention, each node includes an operating system that executes various operating system tasks. The logic for performing the Dijkstra shortest path computation is a component of one particular operating system task that is referred to hereinafter as the "routing task." The routing task implements a link state routing protocol, which involves performing the Dijkstra shortest path computation.

In a typical prior art embodiment, the routing task is executed at a predetermined priority level of the operating system. For convenience, this predetermined priority level is referred to hereinafter as the "low priority level," and an operating system task that operates at the low priority level is referred to hereinafter as a "low priority operating system task." Other operating system tasks operate at the low priority level, and the operating system performs multitasking in order to allow the various low priority operating system tasks, including the routing task, to share processing resources. This multitasking typically allows a low priority operating system task to execute for a predetermined period of time (or less, if the task relinquishes control), after which the operating system performs a context switch to allow another low priority operating system task to execute. Thus, a low priority operating system task, such as the routing task, may be interrupted by the operating system in order to allow another low priority operating system task to execute. Furthermore, a low priority operating system task, such as the routing task, may be interrupted by the operating system in order to allow a higher priority operating system task to execute.

Because the Dijkstra shortest path computation is computationally intensive, the Dijkstra shortest path computation can take a relatively long time to complete, especially if the communication network includes many nodes. Inevitably, the routing task is interrupted (perhaps multiple times) during the Dijkstra shortest path computation. This, in turn, increases the amount of time required to complete the Dijkstra shortest path computation.

Therefore, an embodiment of the present invention raises the routing task to a high priority level in order to perform the Dijkstra shortest path computation, and lowers the routing task back to the low priority level upon completion of the Dijkstra shortest path computation. In this way, the routing task is able to execute without being interrupted by the remaining low priority operating system tasks or other operating system tasks running at a lower priority level than the routing task. This, in turn, enables the routing task to complete the Dijkstra shortest path computation with fewer interruptions, thereby reducing the amount of time required to complete the Dijkstra shortest path computation.

In an embodiment of the present invention, the node includes task priority control logic that controls the priority level of the routing task. The task priority control logic may be a component of the routing task or may be distinct from the routing task. The task priority control logic causes the routing task to execute at the low priority level when the routing task is not performing the Dijkstra shortest path computation and at the high priority level when the routing task is performing the Dijkstra shortest path computation.

More specifically, the task priority control logic causes the routing task to execute at the low priority level as the default priority level for the routing task. The task priority control logic monitors for a condition that triggers the Dijkstra shortest path computation, which, in a preferred embodiment, is the receipt of a LSA protocol message including link status information. Upon detecting the trigger condition, the task priority control logic raises the routing task to the high priority level so that the routing task performs the Dijkstra shortest path computation at the high priority level. Upon completion of the Dijkstra shortest path computation, the task priority control logic lowers the routing task back to the low priority level, and monitors for a subsequent trigger condition.

FIG. 1 is a block diagram showing the relevant components of an exemplary node 100 in accordance with an embodiment of the invention. The node 100 includes, among other things, an operating system 102, a task 104, and a number of network interfaces 110. The task 104 includes, among other things, computation logic 106. The task 104 is associated with task priority control logic 108, which may or may not be an actual component of the task 104. The operating system 102 supports at least a high priority level and a low priority level. The task priority control logic 108 causes the task 104 to operate at the low priority level as the default priority level for the task 104. The task priority control logic 108 raises the task 104 to the high priority level upon detecting a trigger condition in order for the computation logic 106 to perform a computation at the high priority level, and lowers the task 104 back to the low priority level when the computation logic 106 has completed the computation.

In a preferred embodiment of the invention, the task 104 is a routing task that executes a link state routing protocol. The computation logic 106 performs a Dijkstra shortest path computation. The trigger condition is the receipt of a LSA protocol message from one of the number of network interfaces 110. The task priority control logic 108 is coupled to the network interfaces 110 via the interface 109 for detecting the trigger condition. The task priority control logic 108 is coupled to the computation logic 106 via the interface 107 to determine when the computation logic 106 has completed the computation. The task priority control logic 108 is coupled to the operating system 102 via the interface 103 to set the priority level for the task 104.

FIG. 2 is a logic flow diagram showing exemplary task priority control logic 200 for expediting the Dijkstra shortest path computation in accordance with an embodiment of the invention. Beginning at step 202, the logic initializes the routing task at the low priority level, in step 204, and begins monitoring for a LSA protocol message, in step 206. Upon receiving a LSA protocol message, in step 208, the logic raises the routing task to the high priority level, in step 210, and begins monitoring for completion of the Dijkstra shortest path computation, in step 212. The logic continues monitoring for completion of the Dijkstra shortest path computation, in step 212, while the Dijkstra shortest path computation is being performed (NO in step 214). When the Dijkstra shortest path computation is complete (YES in step 214), the logic lowers the routing task back to the low priority level, in step 216, and recycles to step 206 to monitor for a next LSA protocol message.

Although the task priority control logic is preferably used to raise the priority level of the routing task in order to perform the Dijkstra shortest path computation at the high priority level, the task priority control logic can be used more generally to raise the priority level of any task that performs an important operation in order to perform that operation at the high priority level.

FIG. 3 is a logic flow diagram showing exemplary task priority control logic 300 for expediting an operation in accordance with an embodiment of the invention. Beginning at step 302, the logic initializes the task at the low priority level, in step 304, and begins monitoring for the trigger condition, in step 306. Upon detecting the trigger condition, in step 308, the logic raises the task to the high priority level, in step 310, and begins monitoring for completion of operation, in step 312. The logic continues monitoring for completion of the operation, in step 312, while the operation is being performed (NO in step 314). When the operation is complete (YES in step 314), the logic lowers the task back to the low priority level, in step 316, and recycles to step 306 to monitor for a next trigger condition.

It should be noted that the present invention is in no way limited to the location of the task priority control logic 108 within the node 100. The task priority control logic 108 may be a component of the task 104, the network interfaces 110, a protocol stack (not shown) that processes protocol messages received over the network interfaces 110, or other component of the node 100.

In a preferred embodiment of the present invention, predominantly all of the task priority control logic is implemented as a set of computer program instructions that are stored in a computer readable medium and executed by an embedded microprocessor system within a node. Various embodiments of the invention may be implemented in any conventional computer programming language. For example, an embodiment may be implemented in a procedural programming language (*e.g.*, "C") or an object oriented programming language (*e.g.*, "C++"). Alternative embodiments of the invention may be implemented using discrete components, integrated circuitry, programmable logic used in conjunction with a programmable logic device such as a Field Programmable Gate Array (FPGA) or microprocessor, or any other means including any combination thereof.

Alternative embodiments of the invention may be implemented as a computer program product for use with a computer system. Such implementation may include a series of computer instructions fixed either on a tangible medium, such as a computer readable media (*e.g.*, a diskette, CD-ROM, ROM, or fixed disk), or fixed in a computer data signal embodied in a carrier wave that is transmittable to a computer system via a modem or other interface device, such as a communications adapter connected to a network over a medium. The medium may be either a tangible medium (*e.g.*, optical or analog communications lines) or a medium implemented with wireless techniques (*e.g.*, microwave, infrared or other transmission techniques). The series of computer instructions embodies all or part of the functionality previously described herein with respect to the system. Those skilled in the art should appreciate that such computer instructions can be written in a number of programming languages for use with many computer architectures or operating systems. Furthermore, such instructions may be stored in any memory device, such as semiconductor, magnetic, optical or other memory devices, and may be transmitted using any communications technology, such as optical, infrared, microwave, or other transmission technologies. It is expected that such a computer program product may be distributed as a removable medium with accompanying printed or electronic documentation (*e.g.*, shrink wrapped software), preloaded with a computer system (*e.g.*, on system ROM or fixed disk), or distributed from a server or electronic bulletin board over the network (*e.g.*, the Internet or World Wide Web).

The present invention may be embodied in other specific forms without departing from the essence or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive.

I claim:

1. A method for expediting an operation in a computer system, the method comprising:

maintaining an operating system task for performing an operation;

executing the operating system task at a low priority level prior to performing the operation; and

raising the operating system task to a high priority level in order to perform the operation.

2. The method of claim 1, wherein raising the operating system task to the high priority level in order to perform the operation comprises:

detecting a trigger condition indicating that the operation is to be performed; and

raising the operating system task to the high priority level upon detecting the trigger condition.

3. The method of claim 2, wherein the operating system task is a routing task, and wherein the trigger condition comprises receipt of a link state advertisement protocol message including link status information.

4. The method of claim 3, wherein the operation is a Dijkstra shortest path computation utilizing the link status information received in the link state advertisement protocol message.

5. The method of claim 1, further comprising:

lowering the operating system task to the low priority level upon completion of the operation.

6. A device comprising:

an operating system;

an operating system task including logic for performing an operation; and

task priority control logic operably coupled to execute the operating system task at a low priority level prior to performing the operation and raise the operating system task to a high priority level in order to perform the operation.

7. The device of claim 6, wherein the task priority control logic is operably coupled to raise the operating system task to the high priority level upon detecting a trigger condition.

8. The device of claim 7, wherein the operating system task is a routing task, and wherein the trigger condition comprises receipt of a link state advertisement protocol message including link status information.

9. The device of claim 8, wherein the operation is a Dijkstra shortest path computation utilizing the link status information received in the link state advertisement protocol message.

10. The device of claim 6, wherein the task priority control logic is operably coupled to lower the operating system task to the low priority level upon completion of the operation.

11. A program product comprising a computer readable medium having embodied therein a computer program for expediting an operation in a computer system, the computer program comprising:

task priority control logic programmed to execute an operating system task at a low priority level prior to performing the operation and raise the operating system task to a high priority level in order to perform the operation.

12. The program product of claim 11, wherein the task priority control logic is programmed to raise the operating system task to the high priority level upon detecting a trigger condition.

13. The program product of claim 12, wherein the operating system task is a routing task, and wherein trigger condition comprises receipt of a link state advertisement protocol message including link status information.

14. The program product of claim 13, wherein the operation is a Dijkstra shortest path computation utilizing the link status information received in the link state advertisement protocol message.

15. The program product of claim 11, wherein the task priority control logic is programmed to lower the operating system task to the low priority level upon completion of the operation.

ABSTRACT OF THE DISCLOSURE

A technique for expediting an operation in a computer system involves raising the priority level of an operating system task from a low priority level to a high priority level in order to perform the operation at the high priority level and lowering the priority level of the operating system task back to the low priority level upon completion of the operation. Task priority control logic causes the operating system task to execute at the low priority level as a default priority level. The task priority control logic monitors for a trigger condition indicating that the operation is to be performed. Upon detecting the trigger condition, the task priority control logic raises the operating system task to the high priority level, and begins monitoring for completion of the operation. Upon determining that the operation is complete, the task priority control logic lower the operating system task back to the low priority level.

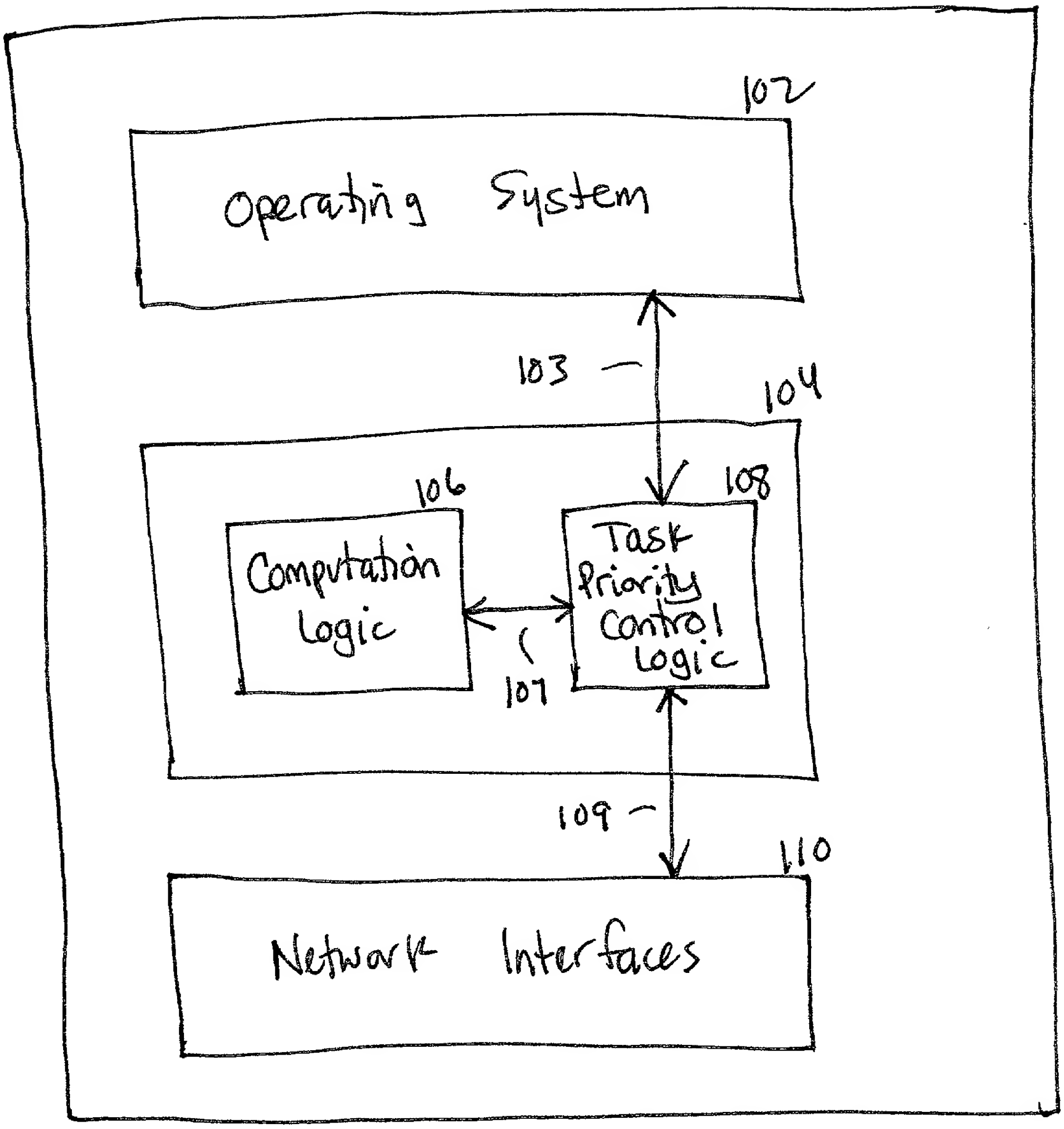
102
Operating System
103
104
106
108
Computation Logic
Task Priority Control Logic
107
109
110
Network Interfaces

FIG. 1 100

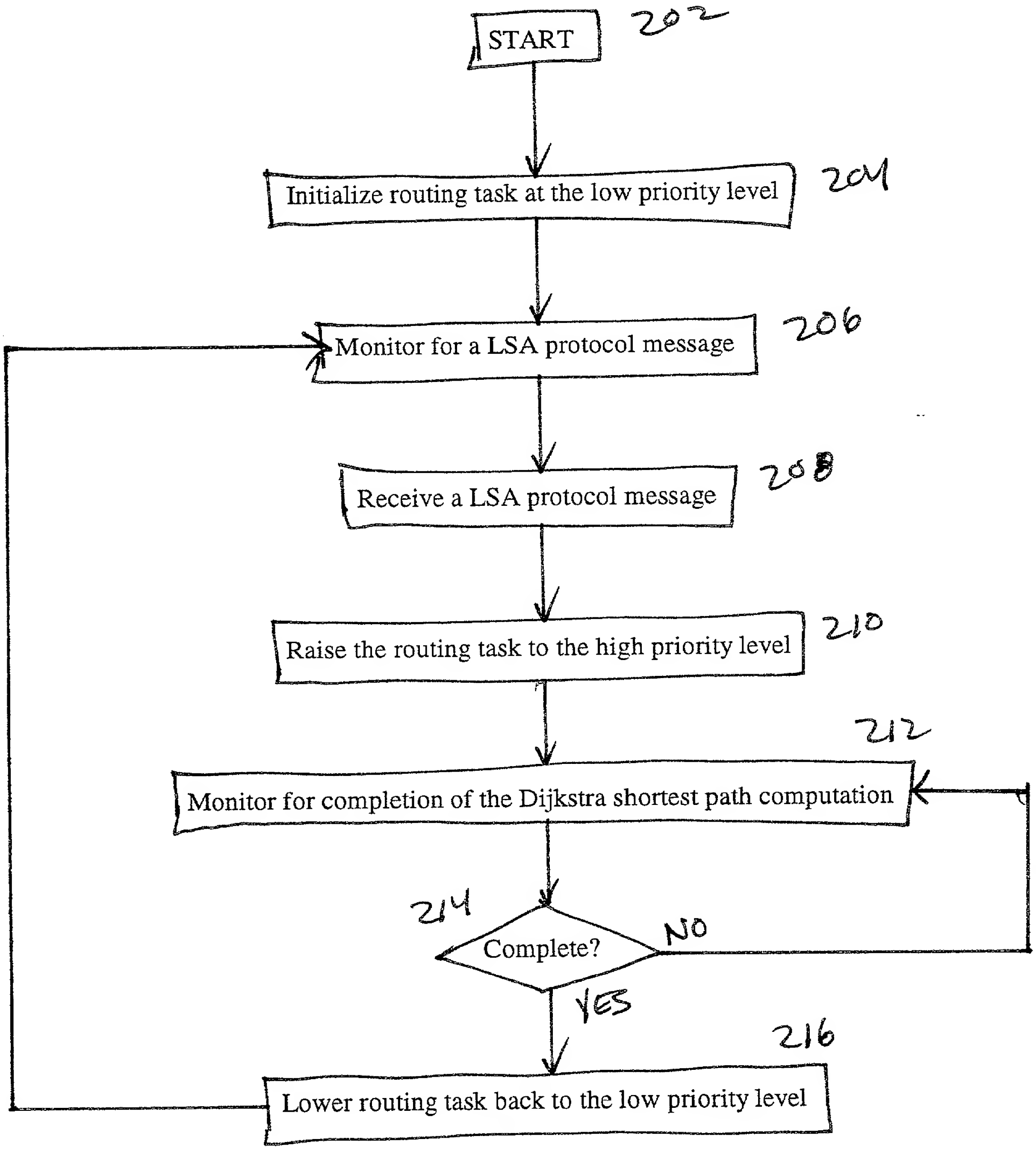
START
202
Initialize routing task at the low priority level
204
Monitor for a LSA protocol message
206
Receive a LSA protocol message
208
Raise the routing task to the high priority level
210
212
Monitor for completion of the Dijkstra shortest path computation
214
Complete?
NO
YES
216
Lower routing task back to the low priority level

FIG. 2 200

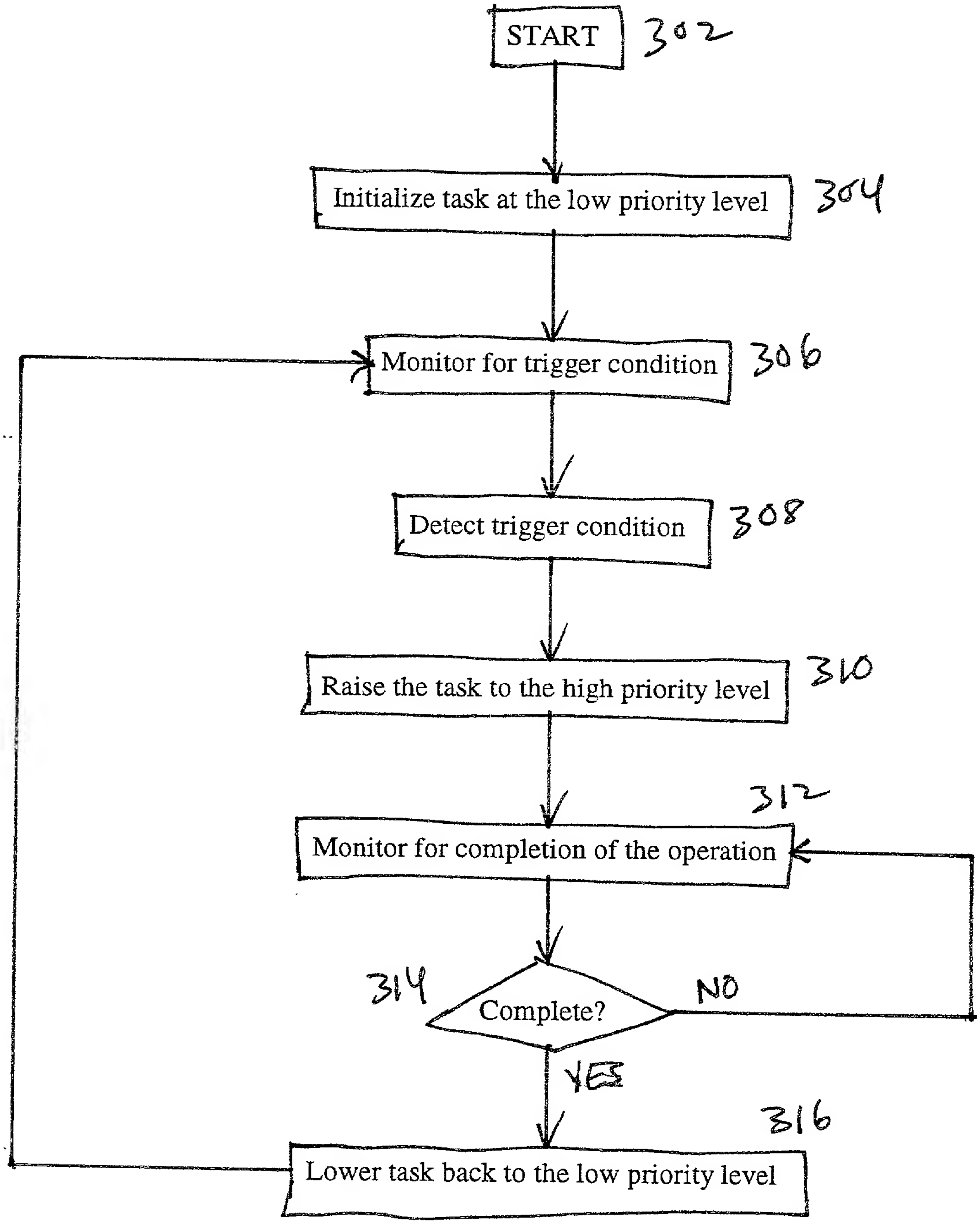
START
302
Initialize task at the low priority level
304
Monitor for trigger condition
306
Detect trigger condition
308
Raise the task to the high priority level
310
312
Monitor for completion of the operation
314
Complete?
NO
YES
316
Lower task back to the low priority level

FIG.3 300

Docket No.
2204/185

Declaration and Power of Attorney For Patent Application

English Language Declaration

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

EXPEDITING AN OPERATION IN A COMPUTER SYSTEM

the specification of which

(check one)

☒ is attached hereto.

□ was filed on ____________ as United States Application No. or PCT International Application Number ____________
and was amended on ____________
(if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d) or Section 365(b) of any foreign application(s) for patent or inventor's certificate, or Section 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate or PCT International application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s)			Priority Not Claimed
(Number)	(Country)	(Day/Month/Year Filed)	□
(Number)	(Country)	(Day/Month/Year Filed)	□
(Number)	(Country)	(Day/Month/Year Filed)	□

I hereby claim the benefit under 35 U.S.C. Section 119(e) of any United States provisional application(s) listed below:

____________________ ____________________
(Application Serial No.) (Filing Date)

____________________ ____________________
(Application Serial No.) (Filing Date)

____________________ ____________________
(Application Serial No.) (Filing Date)

I hereby claim the benefit under 35 U. S. C. Section 120 of any United States application(s), or Section 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of 35 U.S.C. Section 112, I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in Title 37, C. F. R., Section 1.56 which became available between the filing date of the prior application and the national or PCT International filing date of this application:

____________________ ____________________ ____________________
(Application Serial No.) (Filing Date) (Status)
(patented, pending, abandoned)

____________________ ____________________ ____________________
(Application Serial No.) (Filing Date) (Status)
(patented, pending, abandoned)

____________________ ____________________ ____________________
(Application Serial No.) (Filing Date) (Status)
(patented, pending, abandoned)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. *(list name and registration number)*

Bruce D. Sunstein	**Reg. No. 27,234**	**Jay Sandvos**	**Reg. No. 43,900**
Robert M. Asher	**Reg. No. 30,445**	**Sonia K. Guterman**	**Reg. No. 44,729**
Timothy M. Murphy	**Reg. No. 33,198**	**Keith J. Wood**	**Reg. No. 45,235**
Steven G. Saunders	**Reg. No. 36,265**	**Mary M. Steubing**	**Reg. No. 37,946**
Harriet M. Strimpel	**Reg. No. 37,008**	**Christopher J. Cianciolo**	**Reg. No. 42,417**
Samuel J. Petuchowski	**Reg. No. 37,910**	**Lindsay J. McGuinness**	**Reg. No. 38,549**
Jeffrey T. Klayman	**Reg. No. 39,250**		
John J. Stickevers	**Reg. No. 39,387**		
Herbert A. Newborn	**Reg. No. 42,031**		
Elizabeth P. Morano	**Reg. No. 42,904**		
Jean M. Tibbetts	**Reg. No. 43,193**		

Send Correspondence to:
Jeffrey T. Klayman
Bromberg & Sunstein LLP
125 Summer Street
Boston, MA 02110

Direct Telephone Calls to: *(name and telephone number)*
Jeffrey T. Klayman at (617) 443-9292

Full name of sole or first inventor
Bradley Cain

Sole or first inventor's signature   Date

Residence
295 Harvard Street #804, Cambridge, MA 02139

Citizenship
U.S.A.

Post Office Address
Same as residence

Full name of second inventor, if any

Second inventor's signature   Date

Residence

Citizenship

Post Office Address